Efficient Minimization of Routing Cost in Wireless Networks

Dr. Anas Ali Hussien
Computer Engineering Department
College of Engineering
Al-NahrainUniversity
anasali78@yahoo.com

Sarah Yahia Ali
Network Engineering Department
College of Information Engineering
Al-Nahrain University
sarahyahia5@yahoo.com

Abstract:

This paper studies how to select a path with the minimum cost. the networks are characterized by limited computational and power source capabilities. Thus, it is necessary to provide an efficient method for producing routing decisions. Wireless networks do not have any fixed communication infrastructure. For an active connection the end host as well as the intermediate nodes can be mobile. In this work we have examined a wireless network and proposed a group of algorithms in order to find an optimal routing path with a low computational cost, using the special Structure of the connectivity graphs of wireless networks. Dijkstra and Bellman approach can be integrated with many one-to-many shortest path algorithms used in network optimization. The simulation results Show that the Advance Dijkstra Algorithm is the most efficient approach comparing with Bellman approach. The proposed algorithm helps to transport packets to their Destinations by keeping the data transfer power of nodes in the lowest level. The algorithm is exposed to a performance evaluation test via a simulation program that is developed in MATLAB version 7.14.

Keywords:
Wireless network, Dijsktra algorithm, Shortest path, Graph theory.

Journal of Engineering and Development, Vol. 17, No.2, 2013, ISSN 1813- 7822

تحقيق الحد الأدنى الكفوء لتوجيه التكلفةِ في الشبكات اللاسلكي

د.أنس علي حسين
مدرس في قسم هندسة الحاسوب
قسم هندسة الحاسوب
كلية الهندسة
جامعة النهرين

ســارة يحيى علي
طالبة ماجستير
قسم هندسة الشبكات
كلية هندسة المعلومات
جامعة النهرين

الخــلاصة:

هذا البحث يدرس كيفية تحديد المسار مع اقل تكلفة. تتميز الشبكات الاسلكيه بقدرات حسابية و مصادر طاقه محدوده. وبالتالي، فمن الضروري لتوفير نظرية كفوءة لانتاج قرارات المسار بالاعتماد على هذه المميزات. الشبكات اللاسلكية لا تملك اي اتصال لبنية تحتية ثابتة. في الاتصال او الربط النشط يمكن للمستخدم وكذلك لعقد الوسط الناقل ان تكون متحركة.

في هذا العمل ، قد درسنا شبكة لاسلكية وافترضنا مجموعة من الخوارزميات من اجل ان نجد على افضل طريق او مسار مع اقل تكلفة حسابية باستخدام هيكل الرسوم البيانية الخاص للاتصالات اللاسلكية.

نظريات الDijkstra مع ال Bellman-Ford يمكن ان تكون متكاملة مع العديد من الخوارزميات (واحد الى مجموعة) اقصر مسار المستخدم في تحسين الشبكة. نتائج البرمجة تبين ان خوارزمية ال Dijkstra هي الاكثر كفاءة مقارنة بالBellman-Ford.

الخوارزمية المقترحة تساعد على نقل الحزم من البيانات الى مستلميهم بواسطة حفظ قوة نقل حزمة البيانات في اقل مستوى للطاقه.

1. Introduction:

The history of wireless networks started in the 1970s and the interest has been growing ever since. The tremendous growth of personal computers and the handy usage of mobile computers necessitate the need to share the information. The great popularity of Internet services make more people enjoy and depends on the networking applications. However, the Internet is not always available and reliable, and hence it cannot satisfy people's demand for communication at anytime and anywhere **[1].** Anand Srinivas and Eytan Modiano,

developed algorithms for finding minimum energy disjoint paths in an all-wireless network, for both the node and link-disjoint cases. Our major results include a novel polynomial time algorithm that optimally solves the minimum energy 2 link-disjoint paths problem, as well as a polynomial time algorithm for the minimum energy *k* node disjoint paths problem.

Adrian Kosowski and Alfredo Navarra and Cristina M. Pinotti Connectivity problem corresponds to the well-known Minimum Spanning Tree problem in graph theory and The Cheapest Path problem corresponds to the well-known Shortest Path problem in graph theory, in the multi-interface setting this problem is still polynomial solvable, and we point out a simple Dijsktra-based algorithm. Charles B. Ward[a,], Nathan M. Wiegand[b,] Metrics to assess the cost of paths through networks are critical to ensuring the efficiency of network routing. Routing metrics over such networks may be viewed as a class of existing shortest path problems, the formal language contrained path problems. Xiaoqing Zhu and Bernd Girod, it is important that routing algorithms not only yield high aggregated bandwidth, but also minimize network congestion to avoid excessive delay. Janos Levendovszky a, Long Tran-Thanh a,*, Gergely Treplan b, Gabor Kiss a, A reliable path is defined in terms of successful packet transfer to the BS despite the loss links. Neighborhood information is used to compute a simple shortest-path routing based on Dijkstra's algorithm.

2. Network model:

Figure (1) shows the system model that consisting of wireless nodes and links between them. All nodes are fitted with a processor, with power source, and data storage memory that have omnidirectional antennas and can dynamically vary their transmission power. The state of the connectivity between nodes, the transmission cost of every possible link, and the data storage cost per node, for every time slot, are the (Elementary Particles) that form a wireless network. The topology of a network determines the transmission cost, in our case we need to know the topology of the network only if we need to visualize it **[2]**. **Figure 1**, shows the evolving network. The colored circles in the white clouds are the nodes of the network. The white cloud at left represents the state of the network during time slot i and the white cloud at the right represents the state of the network in the next time slot i+1.

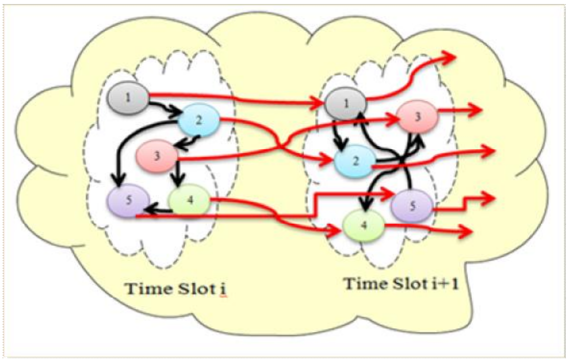

Figure (1) Network model.

The black edges of **Figure 1** represent the available links between the nodes; have a weight that denotes the transmission cost of each link. The red edges indicate the transition cost from one time slot to the next one. The weight of red edges represent data storage cost while moving from the time slot that the red edge begins, to the time slot that the red edge ends **[2].**

In graph theory, the shortest path problem is the problem of finding a path between two vertices (or nodes) in a graph such that the sum of the weights of its constituent edges is minimized**[3].** Shortest path algorithms can be used to find an optimal sequence of choices to reach a certain goal state, or to establish lower bounds on the time needed to reach a given state. The shortest path finding between two nodes in a graph is useful in traveling salesman problem (finding shortest path between two cities), robotics (guiding a robot along a path), routing in wireless-networks (establishing a voice call between two mobile phones), sensor networks (transferring collected data to base-station with minimal battery energy) **[4]**. Graph algorithms are a significant field of interest within computer science. Typical higher-level operations associated with graphs are: finding a path between two nodes, like depth-first search and breadth-first search and finding the shortest path from one node to another, like Dijkstra's algorithm**[5].** Network is a set of nodes that evolve in time, while graph is a set of all nodes consists the network in each time slot, Example if a network has 5 nodes and we are examing 3 periods; the graph has 3x5=15 nodes **[2].** In more detail, we divide the time evolution of wireless network in time slots, where the numbers of nodes, the choice for the routing cost metric, the data storage cost and the connectivity state of every node in a single time slot are given. The computational cost of the advance methods we propose for the optimal cost journeys set in wireless networks is significantly less than that of typical approaches that do not make use of the special structure of the connectivity graphs of wireless networks. The performance of every algorithm is compared to the classic approach of finding the minimum cost path, and a potential use of our algorithms is presented **[2].**

2.1 Weights of edges and links:

The weight of a link can be chosen following a metric according to the nature of the network. It could be chosen considering power control, physical distance, and bandwidth cost. While weight of an edge could depend on the available buffer size of a node, or the amount of time slots a data packet has been stored in that node, or another metric of our choice that models a specific network. We make the assumption that during single time slot, many transmissions can take place between nodes. Between those transmissions there is no data storing cost [2].

3. Dijkstra's algorithm:

Is a graph search algorithm that solves the shortest path problem for a graph with nonnegative edge path costs. This algorithm is often used in routing and as a subroutine in other graph algorithms. The algorithm works by repeatedly selecting a neighbor of current node, that is closest to the destination node. That is, in each iteration/hop, the neighbor closest to destination is added to path. For example, if node n_3 has four neighbors N_5, N_4, N_6, N_7 with distance to destination as 4, 2, 5, 6, and then N_4 is chosen as the next-node in path. Neighbors already visited (added to path) are ignored. The algorithm starts from source-node and repeats itself until the destination-node is reached or no neighbors are found. Neighbors of a node are those that are directly connected to that node. If the graph is un-directed or if there are no edges between nodes (un-connected), then the neighbors are chosen based on domain-specific criteria. In wireless-network routing problem, the neighbor nodes are those that are within the transmission range of current node (mobile phone). The word “journey” with the minimum cost path between two vertices that belong in different time slots also, while “path” we mean the minimum cost path between two nodes in the same time slot **[2].**

4. Bellman-Ford algorithm:

Is a graph search algorithm that can handle edges with negative weight of traversal (better to pass through such an edge than not to pass). Edges with negative weights may arise after transforming many real world problems into graph search algorithm. Bellman-Ford algorithm runs in O (V E) (number of nodes multiplied by the number of vertices) time. Hence it is less efficient than Dijkstra algorithm and only should be used when graph edges may have negative weights (Dijkstra algorithm cannot handle such graphs) **[6].**

5. Advance Approach for Wireless Networks Routing:

In order to develop more efficient approaches for finding routing costs in wireless networks; an advance approach outlined in the following step is presented and implemented:

a) *Create a new graph containing all the nodes and links in a given time slot.*
b) *To calculate the time it required to create the network.*
c) *Add a virtual vertex which represents the initial node of the journey set.*
d) *If the time slot :*

i. *Is the first* one, add a virtual link (**e_i**) beginning *from the virtual vertex and ending to the initial node.*

- *The weight of e_i should be equal to zero.*

ii. *Is not the first one, add n virtual links.*

- *The w_i ($i = 1 \rightarrow n$) of each virtual link will be equal with the sum of journey's cost from the virtual node to node i, plus the storage cost of node i during the previous and the present time slot.*

e) *Find all the journeys from every node to every node that contains output data of shortest paths algorithm for every node.*
f) *Calculated the time of all paths found.*
g) *Execute any generic one-to-many shortest path algorithm (Dijkstra) on the extended graph.*
h) *Plot the journeys from one node to another and Calculated the time that plotting finished.*
i) *Store the journeys found and their costs.*
j) *Calculated the total time to compare each algorithm with another and chose more efficient and therefore less time.*
k) *Repeat steps a to j until all time slots are examined.*

In this paper, implementations of one-to-many algorithms are presented because of its flexibility. We are free to work either solving the problem centralized (pre calculate the full set of optimal journeys from every node to every node of the net centrally), or distributed (each node will find its own optimal journeys to every other node of the net). Advance Dijkstra algorithm is implemented in our model as shown in **Figure 2.**

By adding a virtual vertex (1*) and one virtual link from node (1*) to node 1. The weight of the virtual link is zero as described in the third step of the advance procedure. After calculating the cost of every path in network, we have determined the shortest path to send data packet in least cost to destination node in wireless network.

The observing of the journey of a data packet transmission from node 1 to node 5 in exact four time slots is shown in **figure (3).** During each time slot, the data packet is transmitted from node 1 to node 1 in last slot, forwarded to node 2 and afterwards to node 3 and to the destination node, where it is stored for three time slots to be transmitted forwarded to node 2 and to node3, who delivers it to its final destination node 5.

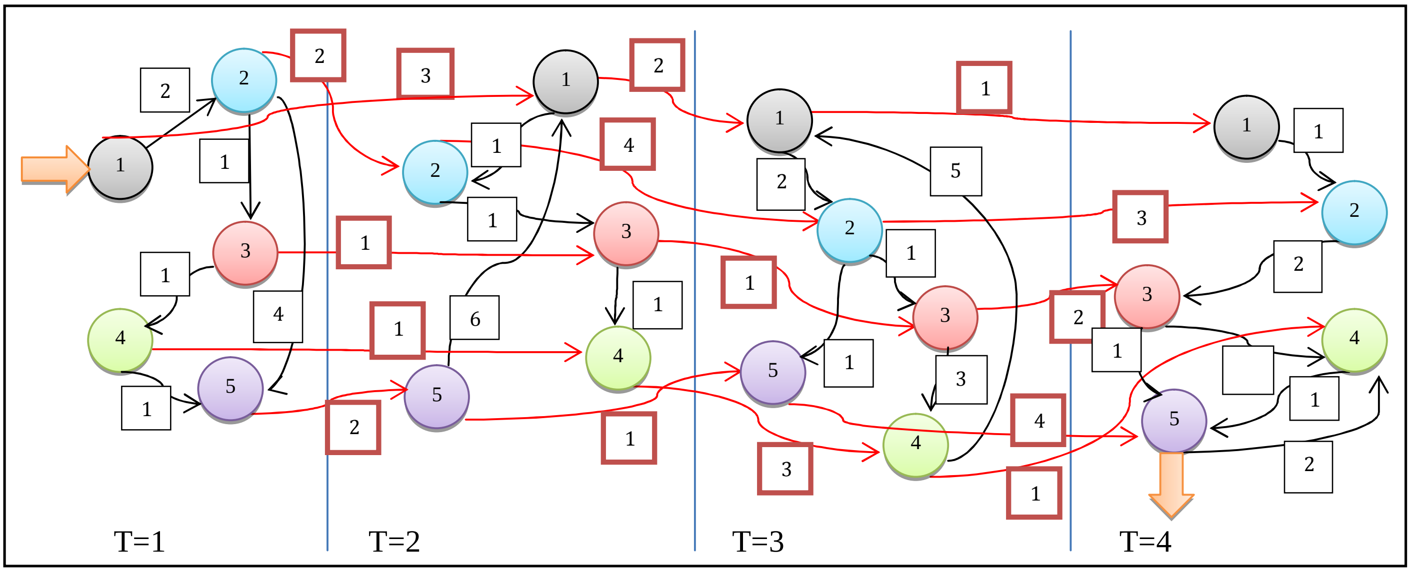

Figure (2) Wireless networks.

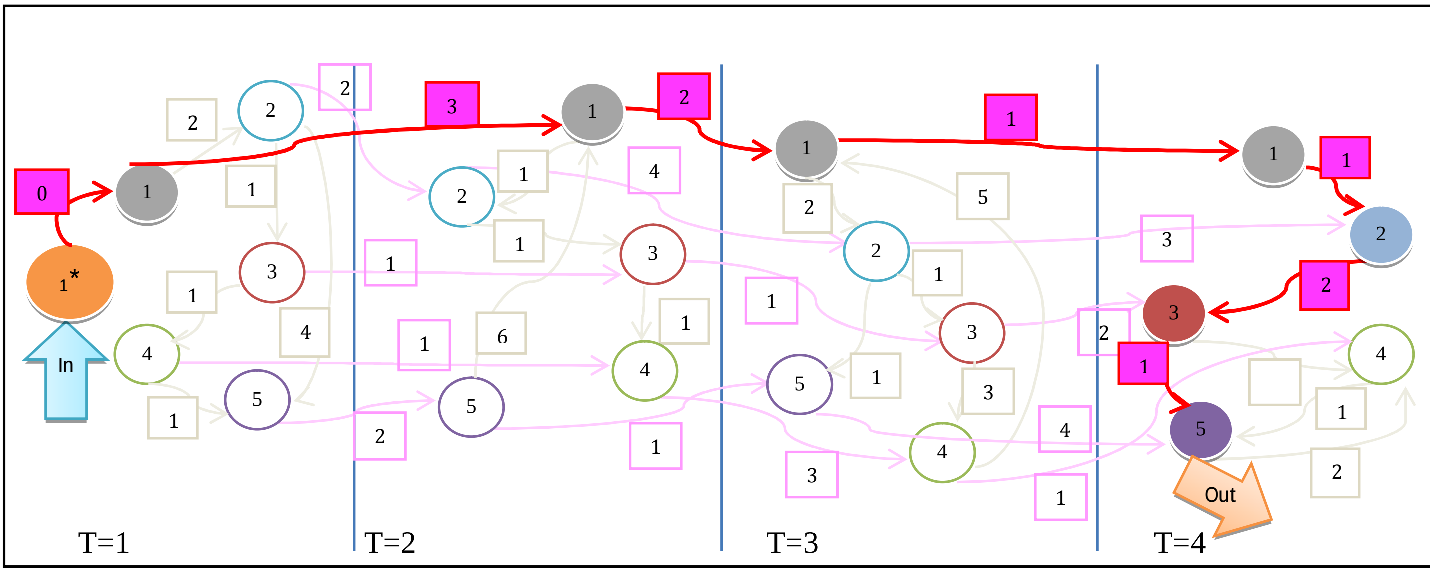

Figure (3) Shortest path.

6. Simulation Results:

Several instances of networks are created with different numbers of nodes and different number of time slots. Then, found the certain time slot for an optimal journey in order to send data packets from every node and deliver it to every node of the network.

Comparison between Advance Dijkstra’s algorithm and Bellman-Ford's is presented. The results prove the demonstrate of the Advance Dijkstra's algorithm over Bellman-Ford algorithms.

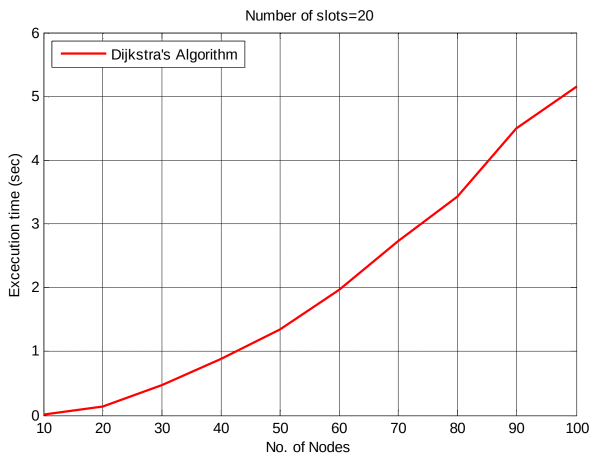

Figure (4) Number of nodes vs. excecution time for Advance Dijkstra’s Algorithm.

Figure 4 shows the execution time in seconds versus the number of nodes for Advance Dijkstra's algorithm, when the number of time slots is constant.While Figure 5 shows the execution time in seconds versus the number of slots for Advance Dijkstra's algorithm, when the number of nodes is 70.

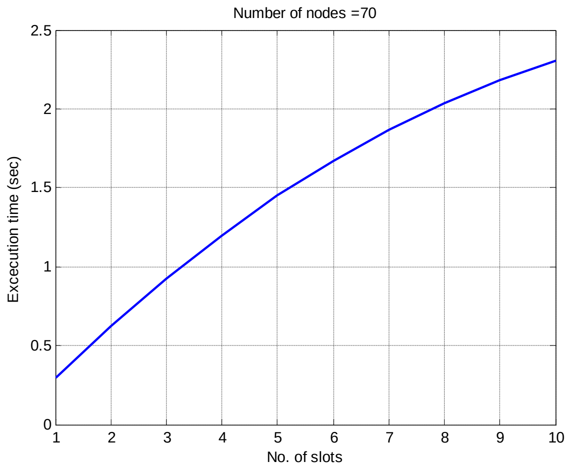

Figure (5) Number of slots vs. excecution time in Advance Dijkstra's Algorithm.

The above simulation is repeated for the Bellman-Ford's Algorithm as shown in **Figure 6** and **Figure 7.**

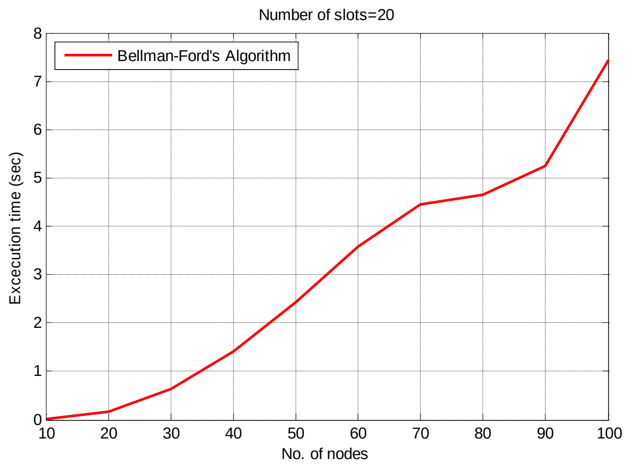

Figure (6) Number of nodes vs. excecution time in Bellman-Ford's Algorithm,

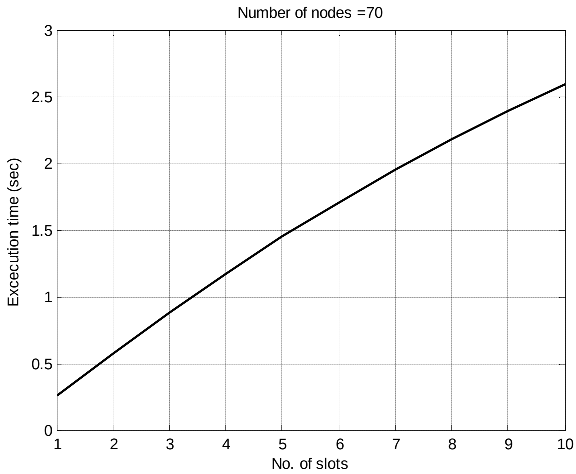

Figure (7) Number of slots vs. excecution time in Bellman-Ford's Algorithm.

By comparing the results **(Figure 8** and **Figure 9**); it's clear that the Advance Dijkstra's Algorithm is more efficient than the Bellman-Ford's Algorithm in wireless network when selecting the shortest path in network routing.

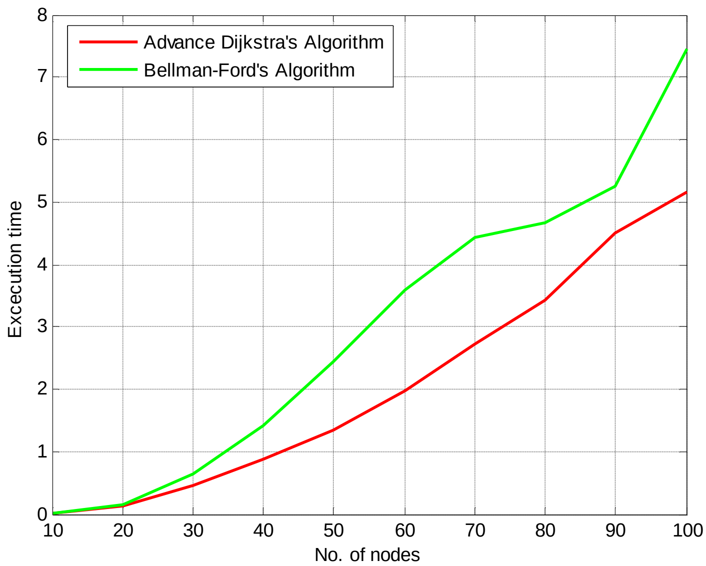

Figure (8) Number of nodes vs. excecution time for Dijkstra'a and Bellman-Ford algorithms.

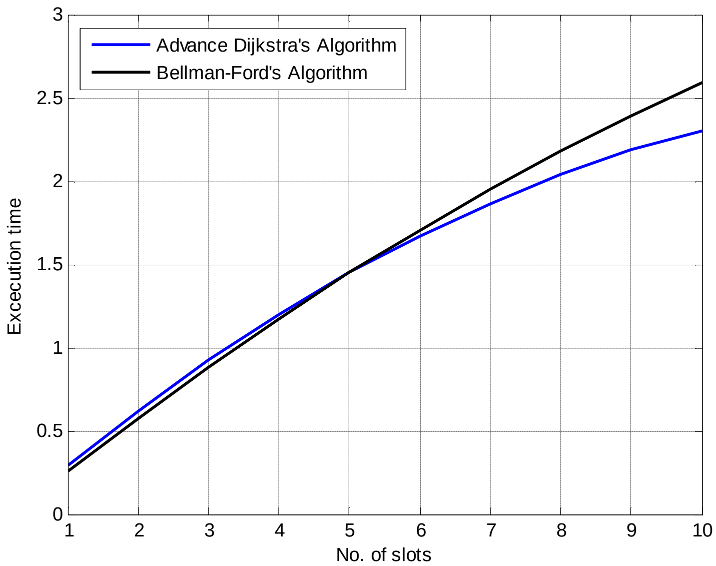

Figure (9) Number of slots vs. excecution time for Dijkstra'a and Bellman-Ford algorithms.

7. Conclusion:

Simulation results shows that Dijkstra algorithm is most efficient approach than Bellman algorithm in terms of transferring packet data in shortest path with lowest cost and higher speed (lowest excecution time).

References:

[1] Ajina A, **"Energy Efficient, Power Aware Routing Algorithm for Sensor Network",** International Journal of Computer Theory and Engineering, Vol.3, No.1, February, 2011.

[2] Christos Tsiaras, **"Efficient Minimization of Routing Cost in Delay Tolerant Networks",** Athens University of Economics and Business, December 2011.

[3]http://www.cs.cornell.edu/courses/cs312/2002sp/lectures/lec20/lec20.htm

[4] Moshe Sniedovich, "**Dijkstra's Algorithm revisited: the OR/MS Connexion**", Department of Mathematics and Statistics, University of Melbourne, 2012.

[5]http://math.fau.edu/locke/GRAPHTHE.HTM.

[6]http://www.richardclegg.org/networks2/Lecture13_06.pdf.

[7] Xiaoqing Zhu And Bernd Girod, "**A Distributed Algorithm For Congestion-Minimized Multi-Path Routing Over Ad Hoc Networks**", Information Systems Laboratory,Stanford University, U.S.A.

[8] Adrian Kosowski Æ Alfredo Navarra Cristina M. Pinotti," **Exploiting multi-interface networks: Connectivity and Cheapest Paths"**, Springer Science+Business Media, LLC 2009.

[9] Janos Levendovszky, Long Tran-Thanh, Gergely Treplan, Gabor Kiss, "**Fading-aware reliable and energy efficient routing in wireless sensor networks",** a Budapest University of Technology and Economics, Department of Telecommunications, Faculty of Information Technology, Peter Pazmany Catholic University, Computer Communications, 2010.

[10] Anand Srinivas and Eytan Modiano, **"Finding Minimum Energy Disjoint Paths in Wireless Ad-Hoc Networks",** Springer Science + Business Media, Inc. Manufactured in the Netherlands, 2005.